

REMARKS/ARGUMENTS

No claims are added, canceled, or amended. Hence, Claims 1-7, 9-20, and 22-30 are pending in the application.

I. SUMMARY OF THE TELEPHONE INTERVIEW

The Examiner and his supervisor are thanked for the telephone interview on October 9, 2008. In the telephone interview, representatives of the Applicants referred to paragraph 48 of the specification as an example of the recited database statement and the recited function. Representatives of the Applicants also discussed the differences between the recited function and the table functions in col. 10 of *Agrawal*. Representatives of the Applicants agreed to consider further amending Claim 1. No further agreement was reached.

II. SUMMARY OF THE REJECTIONS

Claims 1-2, 4-7, 12-15, 17-20, and 25-28 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,324,533 issued to Agrawal et al. ("*Agrawal*") in view of U.S. Publication No. 2002/0087561 to Chen et al. ("*Chen*"). This rejection is respectfully traversed.

Claims 3, 9-11, 16, and 22-24 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Agrawal* in view of *Chen*, and further in view of U.S. Patent No. 6,138,117 issued to Bayardo ("*Bayardo*"). This rejection is respectfully traversed.

Claims 29 and 30 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Agrawal*, in view of *Chen*, in view of *Bayardo*, and further in view of U.S. Patent Publication No. 2002/0059191 to Tamura ("*Tamura*"). This rejection is respectfully traversed.

III. THE REJECTIONS BASED ON THE CITED ART

Claims 1-2, 4-7, 12-15, 17-20 and 25-28 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Agrawal* in view of *Chen*.

A. CLAIM 1

Claim 1 recites:

A method for performing a frequent itemset operation, the method comprising the steps of:
 within a database server that supports a particular database language, parsing a database statement to detect within the database statement, a construct that extends the particular language,
 wherein the construct identifies **a function that counts and returns frequent itemsets** given a cursor as input to the function;
wherein the cursor is used by the function to access values from rows that are returned from a SELECT statement;
wherein the function identifies said frequent itemsets based on said values from said rows returned by said SELECT statement;
 performing said frequent itemset operation as part of execution of the database statement to produce results; and
 storing the results in a computer-readable storage medium. (emphasis added)

As discussed in the telephone interviews on May 6, 2008 and October 9, 2008, Applicants are not claiming to be the first inventors to count and return frequent itemsets. Indeed, paragraphs 7-10 of the Background of the Invention section of the specification concedes, in paragraphs 7 and 9, that frequent itemset operations were known. The Background section further describes the state of the art that existed at the time the present application was filed:

Unfortunately, there is a limit to the type of operations that SQL directly supports. Operations that are not directly supported by SQL may be performed by specifying a series of SQL operations which, when executed in combination with each other, perform the desired unsupported operation.

Depending on the nature of the unsupported operation, the combination of SQL operations required to perform the unsupported operation may be quite complex. Further, amount of time and resources required to execute the series of operations may make the use of SQL impractical.

An example of a type of operation that, in general, cannot be performed efficiently using SQL operations is a frequent itemset operation.

When performed using available SQL operations, frequent itemset operations typically require, among other things, **so many join operations** that performance is frequently unacceptable when the operation involves any sizable item group population. (emphasis added)

Agrawal suffers the same problems identified in the Background of the Invention section. For example, FIGs. 4, 5, 8, 10, and 11 of *Agrawal* depict SQL queries that require numerous join operations and table function calls. Further, col. 6, line 61 to col. 7, line 61 teaches “a process for finding frequent itemsets.” That process includes (a) candidate generation, which requires a pruning step, such as is depicted in FIG. 4, and (b) counting support to find frequent itemsets. The counting support is described in detail in col. 8 line 38 to col. 13, line 67 and depicted in FIGs. 8, 10, and 11. Counting support using SQL-92 shows the use of k-way joins, 3-way joins, subquery-based counting, and multiple group-bys. Counting support using SQL with OR extensions also teaches the use of k-way joins.

In contrast, using the invention recited in Claim 1, in order to count and return frequent itemsets, a user merely has to compose a database statement that references a function that:

- **counts and returns frequent itemsets;**
- **has a cursor as an input parameter;**
- **uses the cursor to access values from rows that are returned from a SELECT statement; and**
- **identifies said frequent itemsets based on said values from said rows returned by said SELECT statement.**

Agrawal lacks any teaching or suggestion of a **function** that satisfies **any** of these limitations, much less **all** of these limitations.

1. Response to “Response to Arguments”

In the “Response to Arguments” section, the Final Office Action points out that “the word ‘single’ [in the above phrase “*Agrawal* lacks any teaching or suggestion of a single

function that counts and returns frequent itemsets”] does not appear anywhere in the claim language.” It is respectfully noted that the term “single” does not have to appear in the claim language because Claim 1 makes it clear that, throughout Claim 1, the Claim is referring to the same function. Specifically, using well-established claiming rules, the function is first introduced as “*a* function” and then consistently referred to thereafter as “**the** function” or “**said** function”.

Also in the “Response to Arguments” section, the Final Office Action equates the group-by-query of *Agrawal* with the recited function of Claim 1. This is incorrect. A group-by query, as is clear on its face, is not a function, but a query.

Finally in the “Response to Arguments” section, the Final Office Action seems to equate the Gather and Comb-K table functions of *Agrawal* with the recited function. This is incorrect. In a nutshell, neither the Gather nor the Comb-K table functions satisfy **any** of the limitations which are explicitly required of the function recited in Claim 1:

- **counts and returns frequent itemsets;**
- **has a cursor as an input parameter;**
- **uses the cursor to access values from rows that are returned from a SELECT statement; and**
- **identifies said frequent itemsets based on said values from said rows returned by said SELECT statement.**

As discussed in the telephone interview of October 9, 2008, the Gather table function of *Agrawal* merely collects all the items of a transaction and outputs a record for the transaction (see col. 10, lines 19-22). Thus, the Gather table function does not count and return frequent itemsets. Therefore, the Gather table function cannot be equated to the recited function of Claim 1. Furthermore, it would not have been obvious to use a cursor in the Gather table function. It is unclear how such a combination would even be possible. The Gather table function only takes two simple inputs: a transaction identifier and an item of the transaction

associated with the transaction identifier. The Gather table function then outputs a record. There is no reason to have the Gather table function take a cursor as input. To attempt to modify the Gather table function to include a cursor as input would destroy the purpose of the Gather table function.

Also as discussed in the telephone interview, the Comb-K table function of *Agrawal* cannot be the recited function of Claim 1. According to col. 10, lines 24-27 of *Agrawal*, Comb-K takes as input the output of the Gather table function and returns all k-item combinations formed out of the items of a single transaction. For example, if $k = 2$ and a transaction contains items A, B, and C, then Comb-K would return {A, B}, {A, C}, and {B, C}. Although Comb-K may return one or more k-item combinations from a transaction, this is far from counting and returning frequent itemsets, as Claim 1 requires. Furthermore, similar in respects to the Gather table function, it would not have been obvious to use a cursor in the Comb-K table function. It is unclear how such a combination would even be possible. The Comb-K table function only takes two simple inputs: a transaction identifier and a list of items of the transaction associated with the transaction identifier. There is no reason to have the Comb-K table function take a cursor as input. To attempt to modify the Comb-K table function to include a cursor as input would destroy the purpose of the Comb-K table function.

Based on the foregoing, *Agrawal* and *Chen* fail to teach or suggest, both individually and in combination, all the features of Claim 1. Therefore, Claim 1 is patentable over *Agrawal* and *Chen*. Reconsideration and withdrawal of the rejection of Claim 1 under 35 U.S.C. § 103(a) is therefore respectfully requested.

B. CLAIMS 2-7, 9-13, 27, AND 29

Claims 2-7, 9-17, 27, and 29 are dependent claims that depend (indirectly or directly) on Claim 1 discussed above. Therefore, each of Claims 2-7, 9-17, 27, and 29 includes the same features of Claim 1 discussed above. Each of Claims 2-7, 9-17, 27, and 29 is therefore patentable over the cited art for at least the same reasons discussed above for Claim 1. Furthermore, each of Claims 2-7, 9-17, 27, and 29 recite additional limitations that may independently render it patentable. Examples follow.

1. *Claim 11*

Claim 11 depends on Claim 3 and additionally recites that additional criteria, specified in the database statement, specifies a set of one or more included items. Claim 11 further recites that performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all itemsets that do not include all items in the set of one or more included items. For example, if the set of included items were A and B, then any frequent itemset that did not include A and B would be excluded from the resulting frequent itemsets.

The Office Action cites col. 3, lines 2-16 of *Agrawal* for allegedly disclosing Claim 11, except for “one or more included items.” By dissecting Claim 11 in this manner, the Final Office Action destroys the meaning of Claim 11. MPEP 2106(II)(c) states that “when evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered.” However, the Final Office Action does precisely this, i.e., the Final Office Action dissects Claim 11 by removing the critical language “one or more include items” from Claim 11 and evaluating that phrase in isolation. Without the phrase “one or more included items”, it is not clear what meaning Claim 11 would have.

In order to render Claim 11 unpatentable over *Agrawal* and *Bayardo*, the Final Office Action must show that such a combination teach or suggest that a database statement specifies one or more included items. And if the one or more included items are not included in an itemset, then that itemset is excluded from the result of performing the frequent itemset operation. However, both *Agrawal* and *Bayardo* fail to teach or suggest that one or more included items are specified in a database statement. The Final Office Action cites col. 3, lines 42-45 of *Bayardo*, which states “(1) generating an initial set C of candidates where each candidate c includes two distinct sets of items: c.head and c.tail.” Neither this cited portion of *Bayardo* nor the cited portion of *Agrawal* teaches or suggests that one or more included items are specified in a database statement. Indeed, both the cited portion of *Agrawal* and the cited portion of *Bayardo* fail to even refer to a database statement.

Furthermore, the cited portion of *Agrawal* is concerned with generating rules from already determined frequent itemsets. Therefore, the cited portion of *Agrawal* is completely unrelated to what the database statement (i.e., that initiates the counting and returning of frequent itemsets) specifies.

2. Claim 29

Claim 29 is similar to Claim 11 in that it depends on Claim 3 and additionally recites what the database statement specifies. Instead of a set of one or more included items, Claim 29 recites a set of one or more excluded items. Itemsets that include all the items in this set are excluded from the results of performing the frequent itemset operation. For example, if the set of excluded items were A and B, then any frequent itemset that did include A and B would be excluded from the resulting frequent itemsets.

The Final Office Action concedes that *Agrawal*, *Chen*, and *Bayardo* does not disclose Claim 29. The Final Office Action then cites paragraph 21 of *Tamura* for allegedly disclosing

Claim 29. This is incorrect. That paragraph merely states that an itemset that includes a combination that is not in a frequent (k-1)-itemset is excluded from the candidate k-itemset.

This paragraph is referring to one stage in the basic algorithm for determining a k-itemset.

Again, this portion mentions nothing about items being specified in a database statement. None of the cited references teach such a database statement, which allows database users to be very flexible in the exact data they want to view.

C. CLAIMS 14-20, 22-26, 28, AND 30

Claims 14-20, 22-26, 28, and 30 are computer-readable storage medium claims that depend on one of the claims discussed above. Each of Claims 14-20, 22-26, 28, and 30 is therefore patentable over the cited art for at least the same reasons discussed above for claim upon which it depends.

IV. CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

/DanielDLedesma#57181/

Daniel D. Ledesma

Reg. No. 57,181

Date: October 10, 2008

2055 Gateway Place, Suite 550

San Jose, CA 95110-1083

Telephone: (408) 414-1080 ext. 229

Facsimile: (408) 414-1076